

Parallele, iterative Löser mit Schur-Komplement-Vorkonditionierung für dünnbesetzte lineare Gleichungssysteme aus der Strömungstechnik

Dr.-Ing. Achim Basermann, Melven Zöllner*



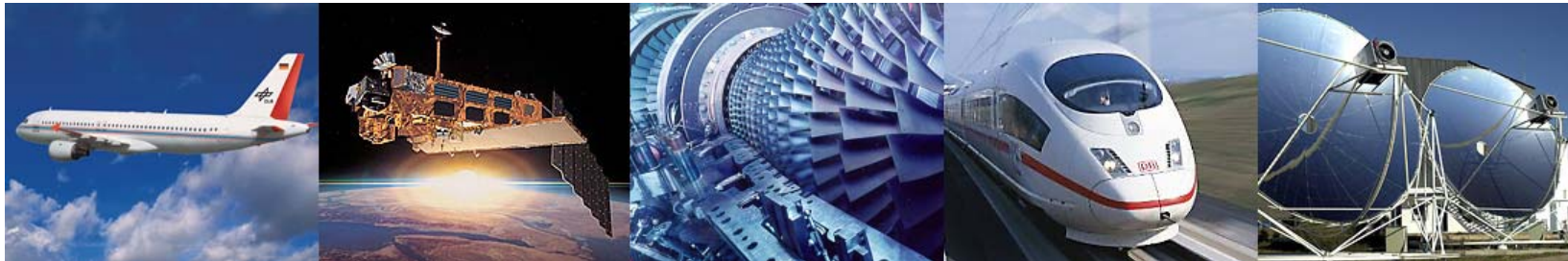
Wissen für Morgen



Übersicht

- Hintergrund DLR
- Anwendungsbeispiel
- Vorkonditionierung
- Ergebnisse
- (Vorkonditionierung bei Eigenwertproblemen)

Das DLR Deutsches Zentrum für Luft- und Raumfahrt



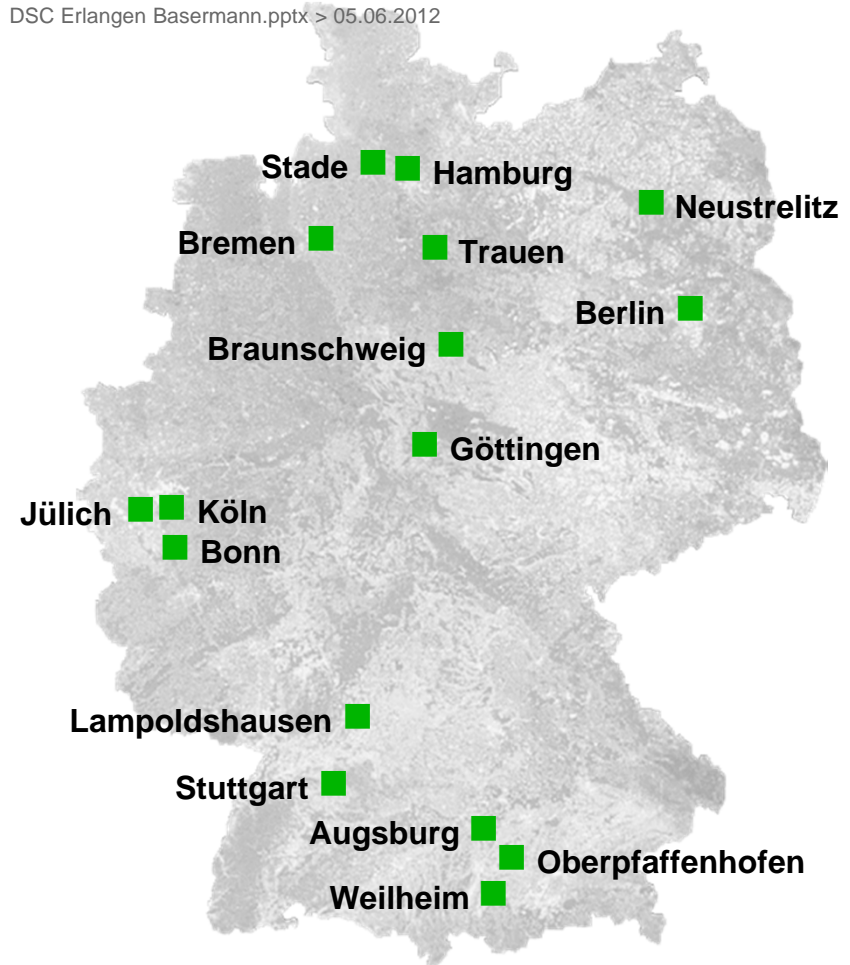
- Forschungseinrichtung
- Raumfahrt-Agentur
- Projektträger



Standorte und Personal

Circa 7.000 Mitarbeiterinnen und Mitarbeiter arbeiten in 32 Instituten und Einrichtungen in
■ 16 Standorten.

Büros in Brüssel,
Paris und Washington.



Forschungsbereiche

- Luftfahrt
- Raumfahrtforschung und -technologie
- Verkehr
- Energie
- Raumfahrtmanagement
- Projektträger



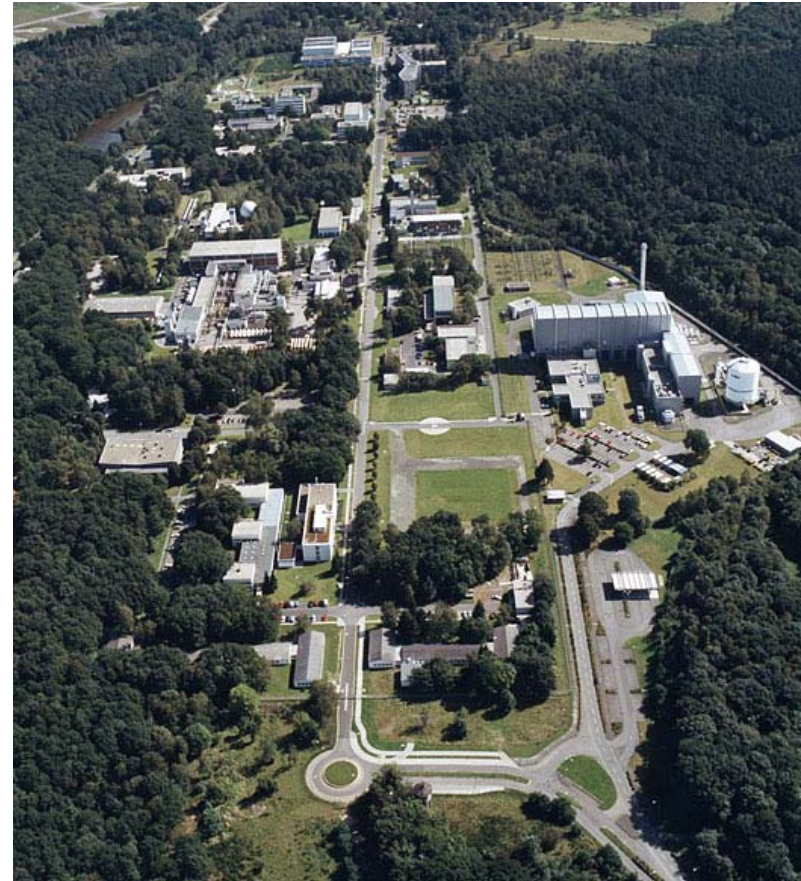
DLR Standort Köln

Mitarbeiterinnen und Mitarbeiter: 1400

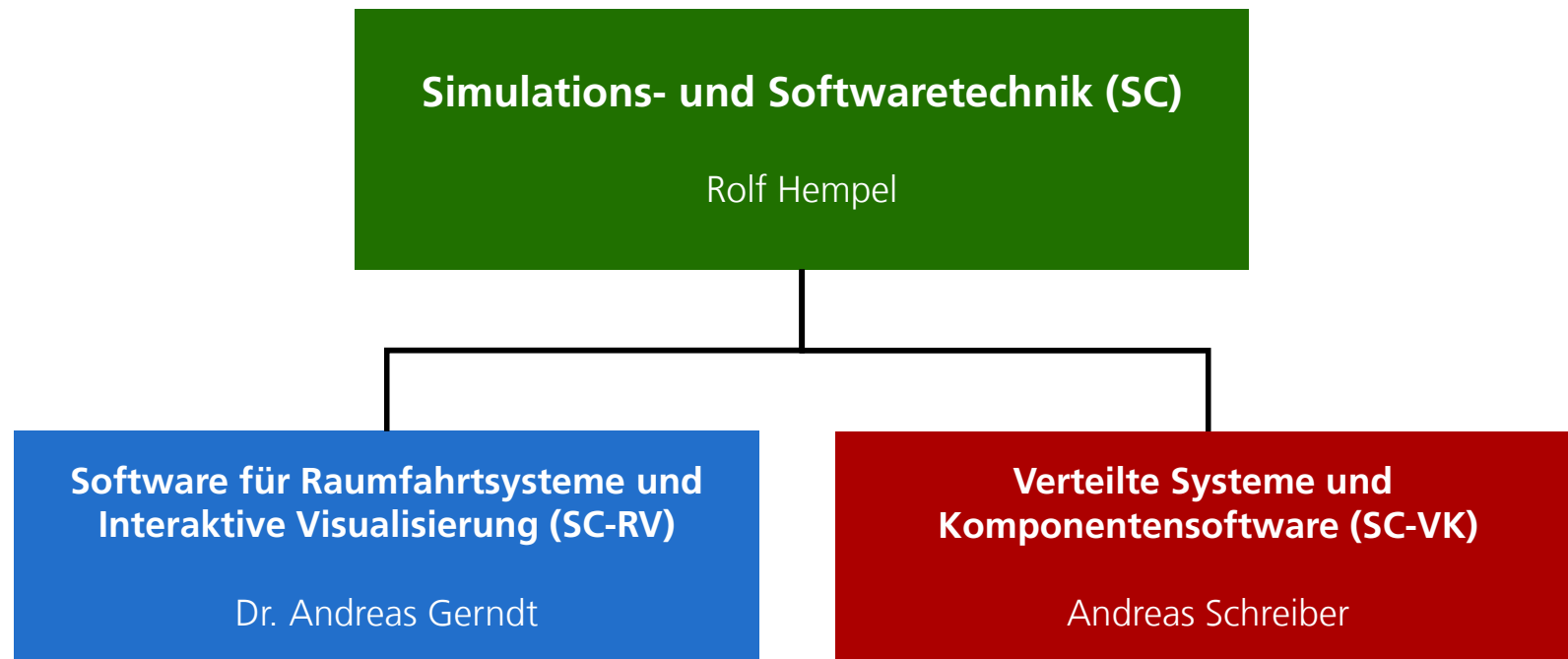
Fläche: 550.000 m²

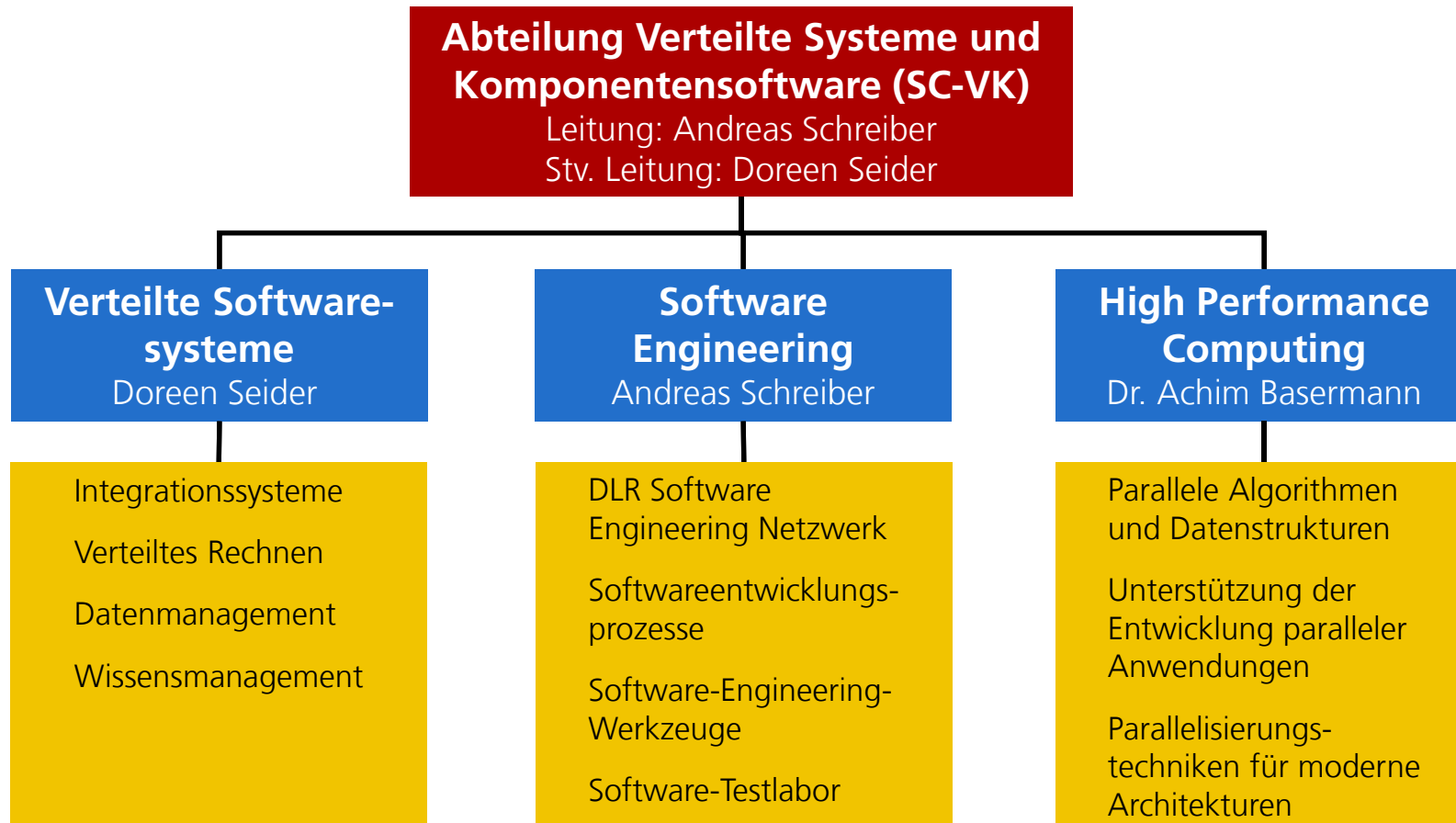
Forschungsinstitute und Einrichtungen:

- Institut für Materialphysik im Weltraum
- Institut für Luft- und Raumfahrtmedizin
- Institut für Antriebstechnik
- Institut für Werkstoff-Forschung
- Institut für Flughafenwesen
- Institut für Solarforschung
- Hochflusssdichte-Sonnenofen
- Simulations- und Softwaretechnik
- Qualitäts- und Produktsicherung
- European Astronaut Center
- Überschall- und Hyperschalltechnologie
- Europäischer Transschall Windkanal
- Nutzerzentrum für Weltraumexperimente MUSC
- Projektträger im DLR



Organigramm Simulations- und Softwaretechnik





Übersicht

- Hintergrund DLR
- Anwendungsbeispiel
 - Simulationssystem TRACE
 - Finite-Volumen-Diskretisierung
 - Strömungsbeispiel
 - linearTRACE
 - Eigenschaften des Gleichungssystems
 - Speichertechniken für dünnbesetzte Matrizen
- Vorkonditionierung
- Ergebnisse
- (Vorkonditionierung bei Eigenwertproblemen)

Simulationssystem TRACE



- ▶ Entwickelt vom Institut für Antriebstechnik des DLR
- ▶ Simulation dreidimensionaler Strömungen in Turbomaschinen
- ▶ Basiert auf kompressiblen, instationären RANS-Gleichungen
- ▶ Diskretisierung mittels Finiter-Volumen



Finite-Volumen-Diskretisierung

RANS-Gleichungen in Erhaltungsform

$$\frac{\partial q}{\partial t} + \operatorname{div} F(q) + S(q) = 0$$

mit Erhaltungsvariablen $q = (\rho \quad \rho U_1 \quad \rho U_2 \quad \rho U_3 \quad \rho E)$,
Flussfunktion $F(q)$ und Quellterm $S(q)$

FV-Diskretisierung

Für ein Kontrollvolumen i gilt:

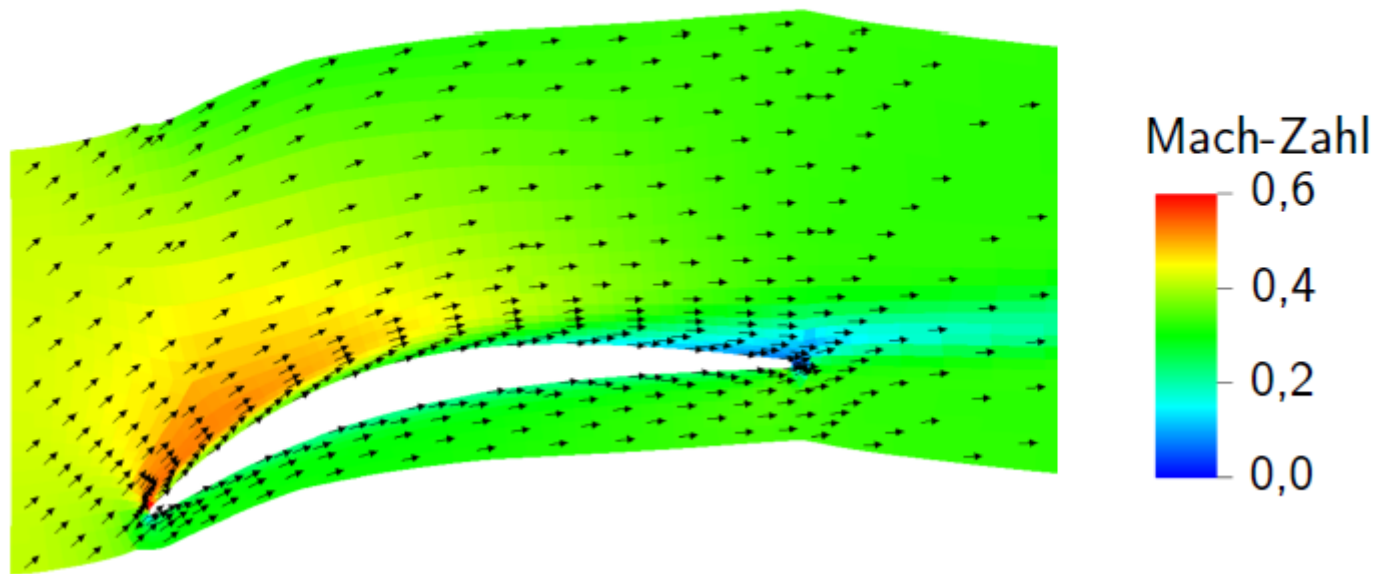
$$\frac{\partial}{\partial t}(V_i q_i) + \sum_{\text{Seiten } j \text{ der Zelle } i} F'_{ij}(\mathbf{q}) + S_i(\mathbf{q})V_i = 0$$



Strömungsbeispiel

TRACE-THD-Testfall

Stationäre gemittelte Strömung in einer Schnittebene



linearTRACE

- ▶ Änderung der Strömung bei Schwingungen der Schaufeln
- ▶ **Eingabe:**
 - ▶ Stationäre Strömungslösung \mathbf{q}^0
 - ▶ Schwingungen des Gitters $\mathbf{x}(t) = \mathbf{x}^0 + \text{Re}(\tilde{\mathbf{x}}e^{i\omega t})$
- ▶ **Ausgabe:**
 - ▶ Strömungsantwort $\mathbf{q}(t) = \mathbf{q}^0 + \text{Re}(\tilde{\mathbf{q}}e^{i\omega t})$
- ▶ Durch Linearisierung der FV-Diskretisierung ergibt sich ein *lineares Gleichungssystem*:

$$\mathbf{A}\tilde{\mathbf{q}} = \mathbf{b} \quad \text{mit} \quad \mathbf{A} \in \mathbb{C}^{(5nz) \times (5nz)}; \mathbf{b}, \tilde{\mathbf{q}} \in \mathbb{C}^{5nz}$$



Eigenschaften des Gleichungssystems

Eigenschaften

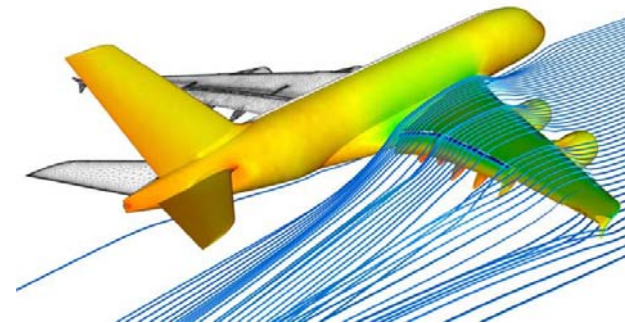
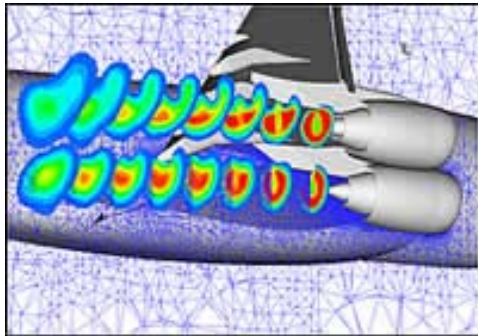
- ▶ jede Zelle nur von einigen umliegenden abhängig
⇒ Gleichungssystem dünnbesetzt
- ▶ pro Zelle 5 Variablen
⇒ dichte 5×5 Blöcke

Übersicht

Testfall	TRACE-UHBR	TAU
Körper	\mathbb{C}	\mathbb{R}
Dimension	4 Mio.	0,5 Mio.
Nichtnulleinträge	620 Mio.	170 Mio.
Blockgröße	5×5	5×5



Simulationssystem TAU: Hintergrund



- TAU: entwickelt für das aerodynamische Flugzeug-Design vom DLR-Institut für Aerodynamik und Strömungstechnik
- Unstrukturierter RANS-Löser (Reynolds-averaged Navier-Stokes), verwendet die Finite-Volumen-Methode
- Erfordert die parallele, iterative Lösung großer, dünnbesetzter, reeller, nicht-symmetrischer, linearer Gleichungssysteme



Speichertechniken für dünnbesetzte Matrizen

Compressed Row Storage (CSR) und Block Compressed Row Storage (BCSR)

Nichtnull-Elemente, zeilenweise:

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

Matrix:

1	0	0	2	0	0
0	3	4	5	0	0
0	0	0	0	6	7
0	0	0	0	8	9

Spaltenindizes, zeilenweise:

1	4	2	3	4	5	6	5	6
---	---	---	---	---	---	---	---	---

Zeiger auf den Zeilenbeginn:

1	3	6	8	10
---	---	---	---	----

1	0	0	2	0	0
0	3	4	5	0	0
0	0	0	0	6	7
0	0	0	0	8	9

- TRACE und TAU verwenden BCSR mit 5x5-Blöcken.
- Vorteil: **weniger indirekte Adressierung**
- Nachteil: **Einige Null-Elemente werden gespeichert.**



Übersicht

- Hintergrund DLR
- Anwendungsbeispiel
- Vorkonditionierung
 - Grundlagen
 - Unvollständige LU-Zerlegungen
 - Block-Jacobi-ILU-Vorkonditionierung
 - Schur-Komplement-Transformation
 - Verteiltes Gleichungssystem
 - DSC-Vorkonditionierung
- Ergebnisse
- (Vorkonditionierung bei Eigenwertproblemen)

Grundlagen

Iterative Verfahren

- ▶ **Problem:** Bestimme x mit $Ax = b$, ($A, \in \mathbb{C}^{n \times n}$, $x, b \in \mathbb{C}^n$).
- ▶ **Ansatz:** verbessere eine Schätzung x_i mit Operator K^{-1} :
 $x_{i+1} := K^{-1}r_i$ mit $r_i := b - Ax_i$
- ▶ Schätzung liegt im Krylov-Raum
 $x_{i+1} \in \mathcal{K}^{i+1} = \{r_0, K^{-1}Ar_0, \dots, (K^{-1}A)^i r_0\}$
- ▶ **Krylov-Unterraum-Verfahren:**
Bestimme geeignete Lösung aus \mathcal{K}^{i+1}
- ▶ Operator K^{-1} wird als **Vorkonditionierung** bezeichnet
 - ▶ beeinflusst Konvergenzverhalten des Verfahrens
 - ▶ Ziel: Minimiere Gesamtaufwand



Unvollständige LU-Zerlegungen (1)

Gauß-Elimination

```
1: for  $i = 1, \dots, n$  do
2:   for  $k = 1, \dots, i - 1$  do
3:      $a_{i,k} \leftarrow a_{i,k} (a_{k,k})^{-1}$ 
4:     for  $j = k + 1, \dots, n$  do
5:        $a_{i,j} \leftarrow a_{i,j} - a_{i,k} a_{k,j}$ 
6:     end for
7:   end for
8: end for
```

- **Block-Verfahren** verwenden Submatrizen, z.B. $a_{i,j} \in \mathbb{C}^{5 \times 5}$



Unvollständige LU-Zerlegungen (1)

Unvollständige Gauß-Elimination

```
1: for  $i = 1, \dots, n$  do
2:   for  $k = 1, \dots, i - 1 \wedge (i, k) \in M$  do
3:      $a_{i,k} \leftarrow a_{i,k} (a_{k,k})^{-1}$ 
4:     for  $j = k + 1, \dots, n \wedge (i, j) \in M$  do
5:        $a_{i,j} \leftarrow a_{i,j} - a_{i,k} a_{k,j}$ 
6:     end for
7:   end for
8: end for
```

- ▶ **Block-Verfahren** verwenden Submatrizen, z.B. $a_{i,j} \in \mathbb{C}^{5 \times 5}$
- ▶ Die Menge M bestimmt, welche Einträge **verworfen** werden.



Unvollständige LU-Zerlegungen (2)

LU-Zerlegung

- ▶ Gauß-Elimination führt zu einer Zerlegung in untere und obere Dreiecksmatrizen $LU = A$.
- ▶ L und U sind in der Regel deutlich **dichter besetzt** als A .
- ▶ Die Lösung x von $LUx = b$ ermittelt man durch
 1. Vorwärtseinsetzen: $y = L^{-1}b$ und
 2. Rückwärtseinsetzen: $x = U^{-1}y$.

Schlechte Parallelisierbarkeit

- ▶ Gauß-Elimination und Vorwärts-/Rückwärtseinsetzen sind schwierig zu parallelisieren.



Block-Jacobi-ILU-Vorkonditionierung

$$K^{-1} = \begin{pmatrix} (\tilde{L}_1 \tilde{U}_1) & & \\ & (\tilde{L}_2 \tilde{U}_2) & \\ & & (\tilde{L}_3 \tilde{U}_3) \end{pmatrix}^{-1}$$

- ▶ Idee: LU-Zerlegungen der Diagonalblöcke können unabhängig voneinander berechnet werden
⇒ effizient parallel durchführbar
- ▶ Nachteil: Einträge außerhalb der Diagonalblöcke werden vernachlässigt



Schur-Komplement-Transformation

$$\begin{aligned} \begin{pmatrix} D & E \\ F & G \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} f \\ g \end{pmatrix} \\ \Leftrightarrow \begin{pmatrix} D & E \\ & S \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} f \\ g - FD^{-1}f \end{pmatrix} \end{aligned}$$

Lösungsverfahren

1. Transformation:
2. Löse Schur-Komplement-System:
3. Rücktransformation:

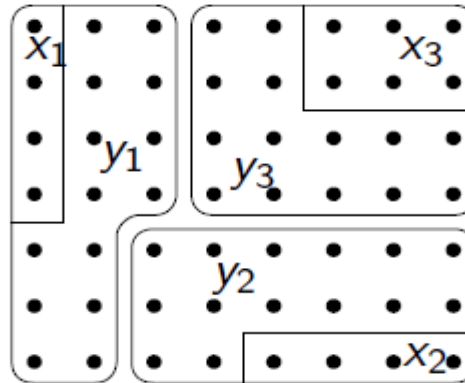
$$g' := g - FD^{-1}f$$

$$Sy = g'$$

$$x = D^{-1}(f - Ey)$$



Verteiltes Gleichungssystem



$$\begin{pmatrix} \begin{matrix} D_1 E_1 \\ F_1 G_1 \end{matrix} & & & & & \\ & G_{1,2} & & G_{1,3} & & \\ & & \begin{matrix} D_2 E_2 \\ F_2 G_2 \end{matrix} & & & \\ G_{2,3} & & & G_{2,3} & & \\ & & & & \begin{matrix} D_3 E_3 \\ F_3 G_3 \end{matrix} & \\ G_{2,3} & G_{3,2} & & & & \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ g_1 \\ f_2 \\ g_2 \\ f_3 \\ g_3 \end{pmatrix}$$



DSC-Vorkonditionierung (1)

- ▶ Schur-Komplement-Transformation für verteiltes Gleichungssystem (DSC, *distributed Schur-complement*)
- ▶ Iterative Verbesserung der Block-Jacobi-ILU-Vorkond.:
Einträge außerhalb der Diagonalblöcke werden berücksichtigt.



DSC-Vorkonditionierung (2)

Algorithmus

1. ILU der Diagonalblöcke:

$$\begin{pmatrix} \widetilde{L}_{D_i} & \\ (F_i U_{D_i}^{-1}) & \widetilde{L}_{S_i} \end{pmatrix} \begin{pmatrix} \widetilde{U}_{D_i} & \widetilde{(L_{D_i}^{-1} E_i)} \\ & \widetilde{U}_{S_i} \end{pmatrix} = \begin{pmatrix} D_i & E_i \\ F_i & G_i \end{pmatrix}$$

2. Transformation:

$$\begin{pmatrix} f'_i \\ y_i^0 \end{pmatrix} = \begin{pmatrix} \widetilde{L}_{D_i} & \\ (F_i U_{D_i}^{-1}) & \widetilde{L}_{S_i} \widetilde{U}_{S_i} \end{pmatrix}^{-1} \begin{pmatrix} f_i \\ g_i \end{pmatrix}$$

3. Approximiere Lösung des **Schur-Komplement-Systems**:

$$y_i + (\widetilde{L}_{S_i} \widetilde{U}_{S_i})^{-1} \sum_{j=1}^3 G_{i,j} y_j = y_i^0 \quad \text{für } i = 1, 2, 3$$

4. Rücktransformation:

$$\begin{pmatrix} x_i^k \\ y_i^k \end{pmatrix} = \begin{pmatrix} \widetilde{U}_{D_i} & \widetilde{(L_{D_i}^{-1} E_i)} \\ & I \end{pmatrix}^{-1} \begin{pmatrix} f'_i \\ y_i^k \end{pmatrix}$$



Übersicht

- Hintergrund DLR
- Anwendungsbeispiel
- Vorkonditionierung
- Ergebnisse
 - Ablauf des Lösungsverfahrens
 - Testsystem
 - Block-Verfahren
 - Parallele Skalierbarkeit
 - linearTRACE-Performance
- (Vorkonditionierung bei Eigenwertproblemen)

Ablauf des Lösungsverfahrens

Lösungsverfahren

1. Berechnen der ILU der Diagonalblöcke
2. Iterative Lösung des Gleichungssystems (FGMRES):
In jedem Schritt:
 - ▶ Vorkonditionierung
 - ▶ Matrix-Vektor-Multiplikation
 - ▶ Vektor-Operationen

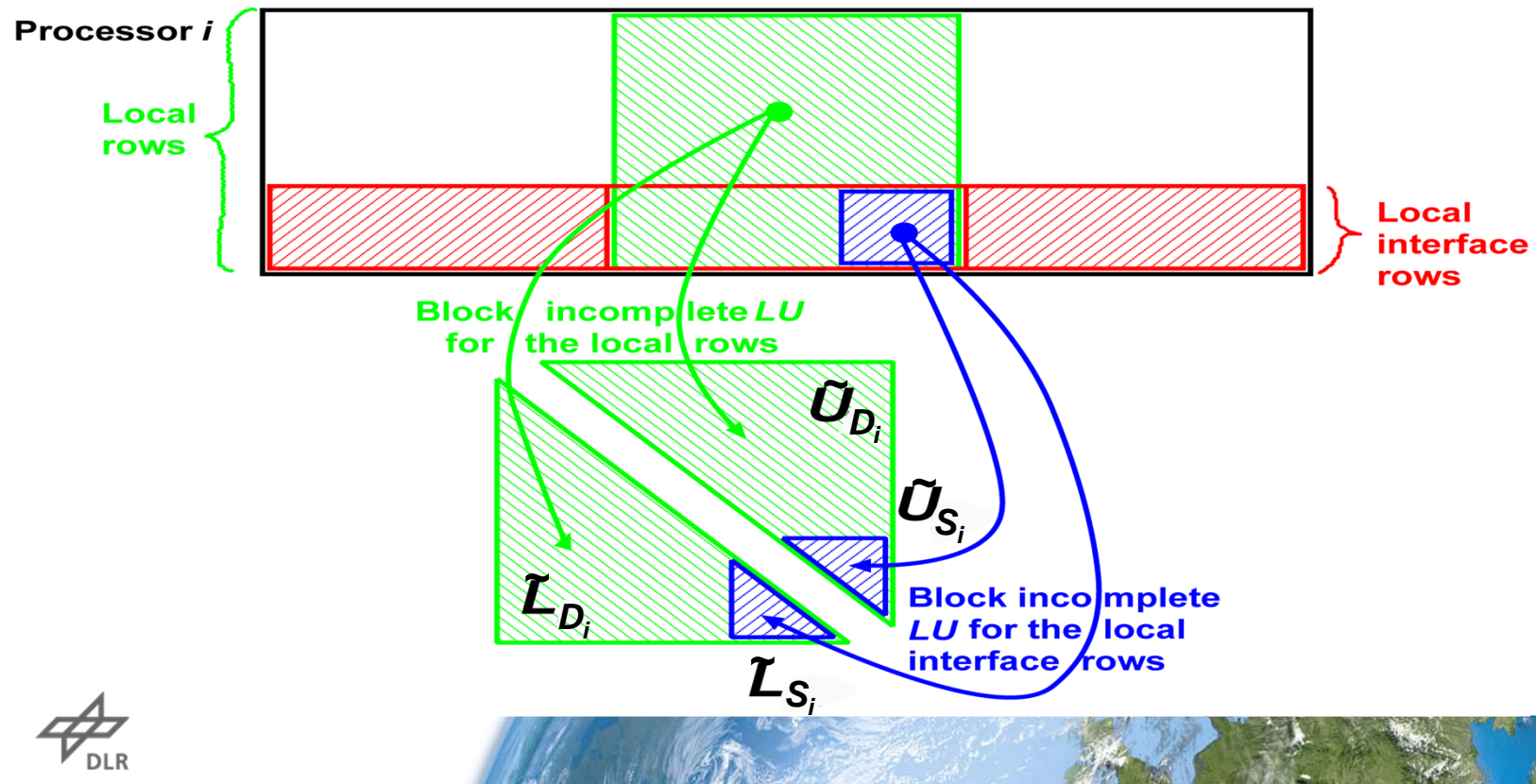
Vorkonditionierung

1. Transformation auf (approx.) Schur-Komplement-System
2. Iterative Approximation der Lösung (GMRES)
3. Transformation auf Ursprungssystem



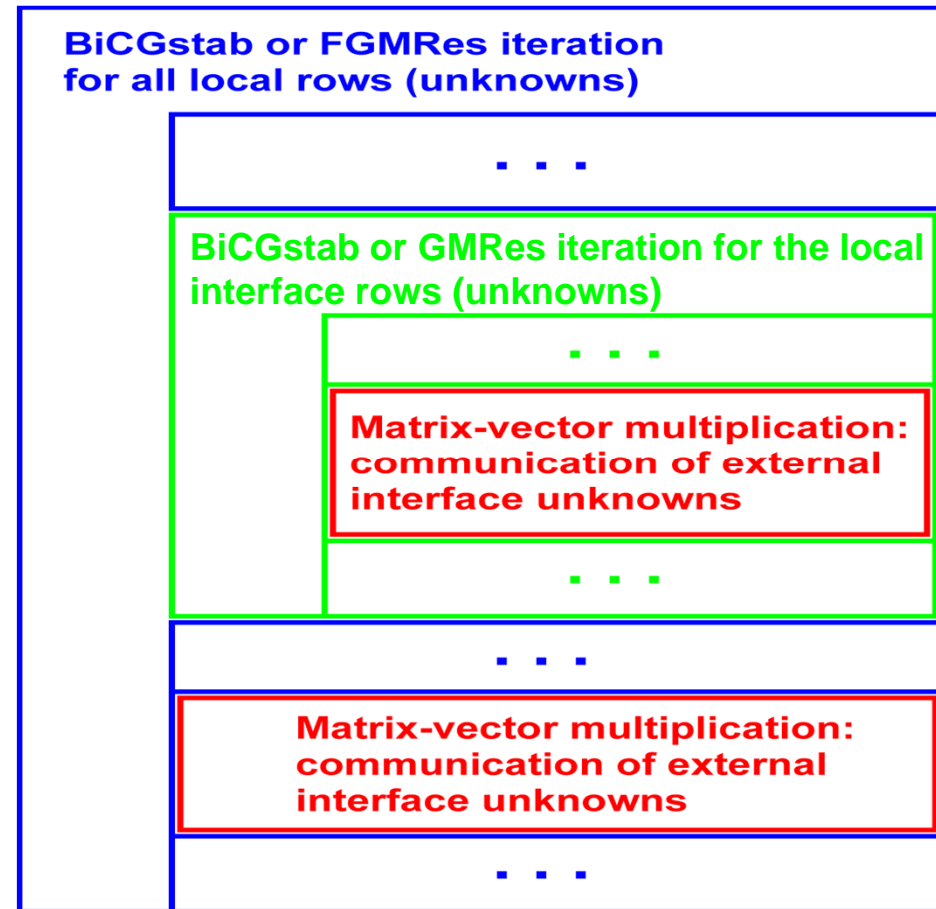
ILU-Berechnung

Preconditioning within the DSC algorithm



Äußere und innere Iteration

Schematic view on
each processor



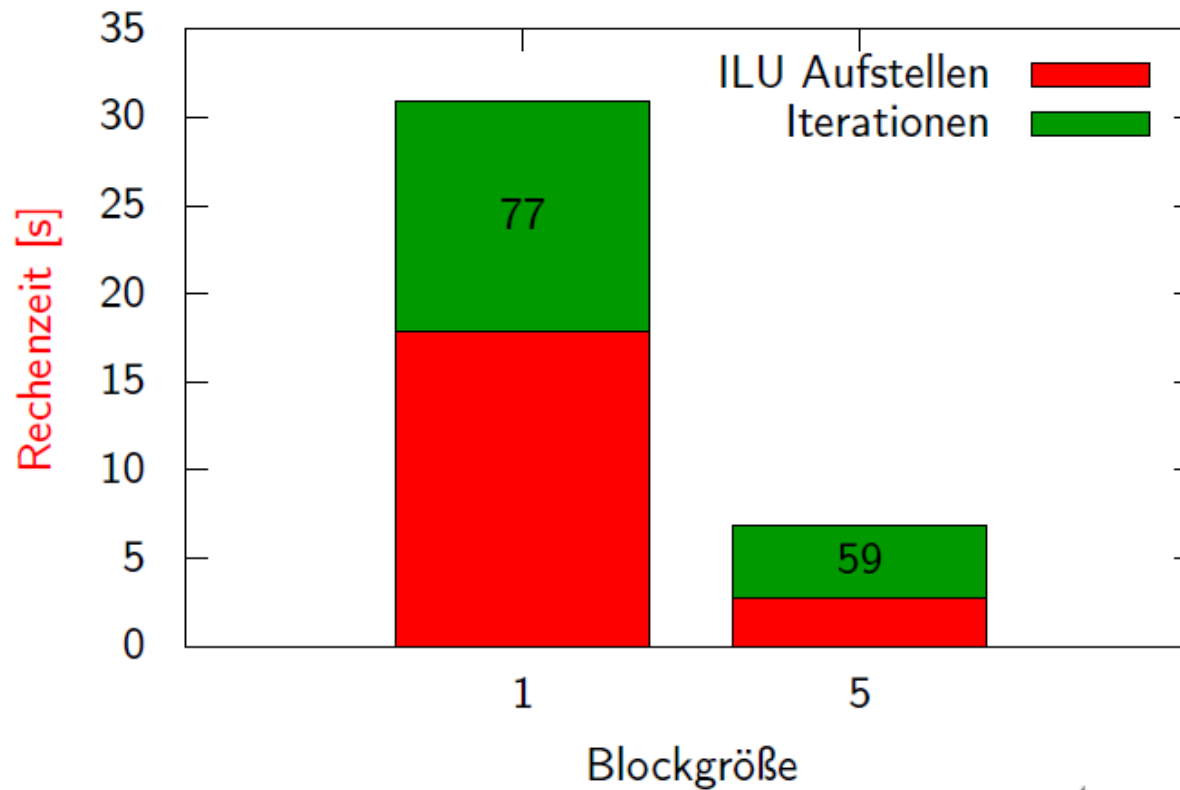
Testsystem

- ▶ **RWTH Bull HPC-Cluster**
 - ▶ Intel Westmere X5675 CPUs
 - ▶ Pro CPU 6 Kerne mit 3,06GHz
 - ▶ 12 Kerne (2 CPUs) pro Node
- ▶ Rechnung mit jeweils einem MPI-Prozess pro Kern



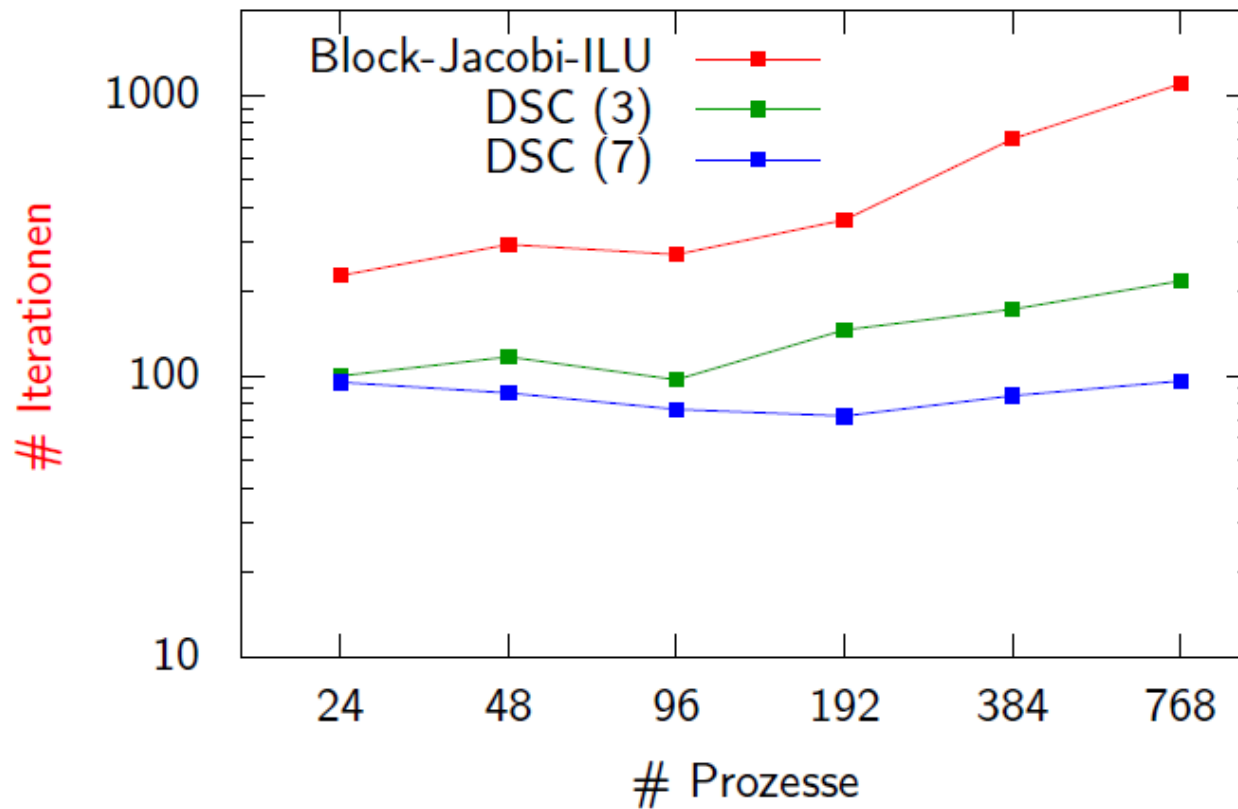
Block-Verfahren

TAU-Testfall, Block-Jacobi-ILU-Vorkonditionierung auf 12 Prozessen



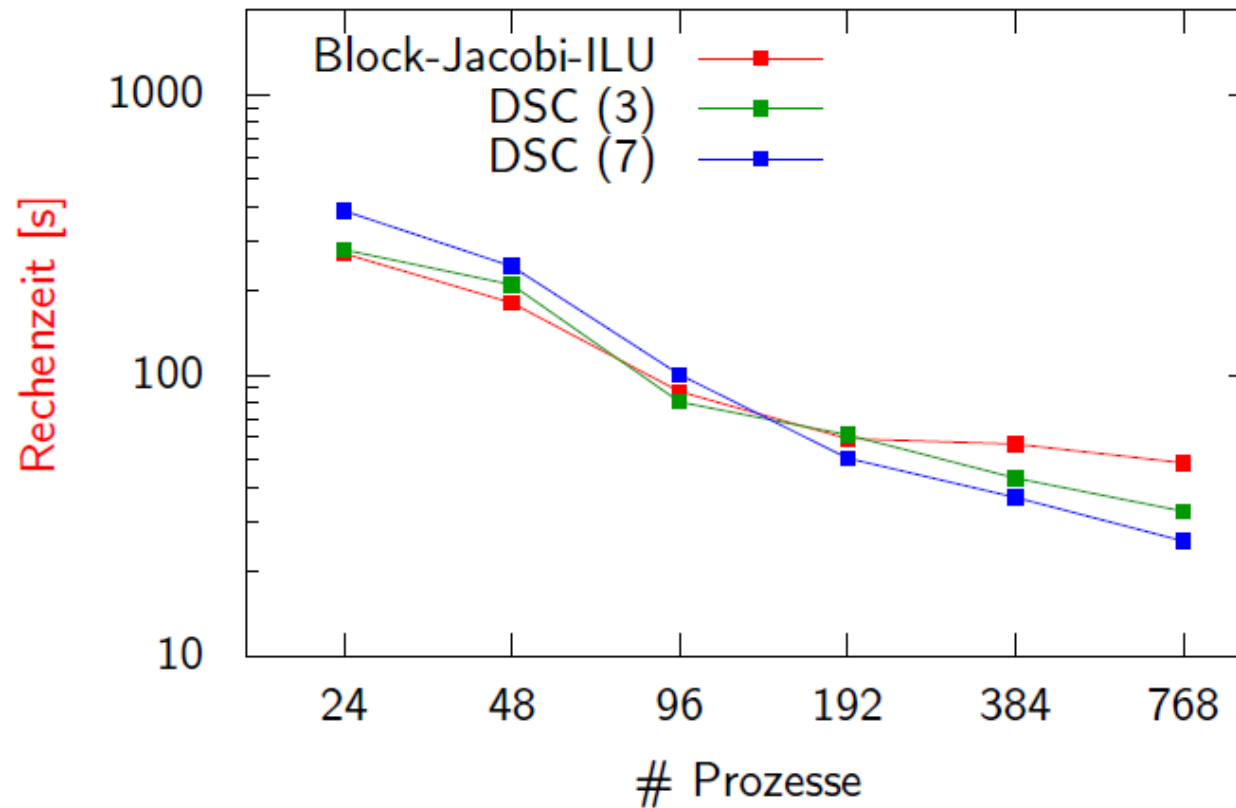
Parallele Skalierbarkeit (1)

TRACE-UHBR-Testfall

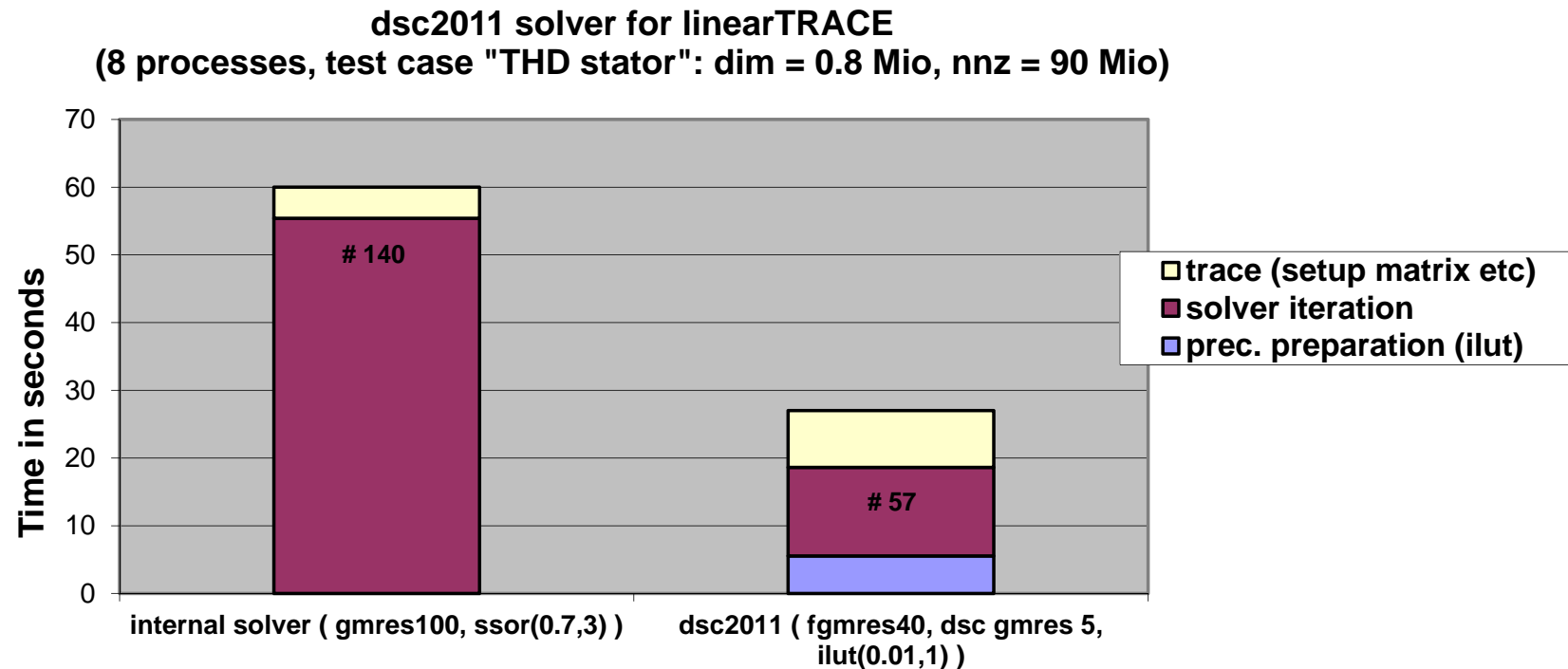


Parallele Skalierbarkeit (2)

TRACE-UHBR-Testfall



linearTRACE-Performance: Interner versus DSC-Löser (2x Intel XEON E5520 mit 4 Kernen, 2.26 GHz)



Zusammenfassung

Fazit

- ▶ **Blockverfahren** verbessern die *Performance*.
- ▶ **DSC-Methoden** verbessern die *parallele Skalierbarkeit*.

Ausblick

- ▶ **Hybride Parallelisierung** könnte die *parallele Skalierbarkeit* weiter verbessern.



Fragen?



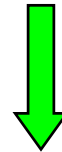
Übersicht

- Hintergrund DLR
- Anwendungsbeispiel
- Vorkonditionierung
- Ergebnisse
- Vorkonditionierung bei Eigenwertproblemen
 - Anwendungen
 - Algorithmen
 - Vorkonditionierung

Eigenwert-Löser (I)

Anwendungen

- Festkörperphysik (Elektronen-Phononen-Wechselwirkung)
- Fernerkundung (Radar)
- Strukturmechanik



Reell symmetrische oder komplex hermitesche Eigenwertprobleme

$$Ax = \lambda x$$



Eigenwert-Löser (II)

Algorithmen

- **Lanczos-Verfahren**
 - Transformation auf tridiagonale Eigenwertprobleme
 - Eigenvektoren über inverse Iteration
 - Langsame Konvergenz
- **Jacobi-Davidson-Verfahren**
 - Extreme Eigenwerte und -vektoren
- **FEAST-Algorithmus**
 - Innere Eigenwerte und -vektoren



Eigenwert-Löser (III): Vorkonditionierung

„Shift and Invert“-Vorkonditionierung

- Transformiere $Ax = \lambda x$ in $(A - \sigma I)^{-1}x = \mu x$
- Bei geeignetem „Shift“ σ Eigenwerte μ_i besser separiert
 - \rightarrow „Vorkonditionierung“
 - \rightarrow schnellere Konvergenz, z.B. bei Arnoldi-Verfahren
- Die Eigenvektoren sind dieselben $\rightarrow \lambda_i$
- Erforderlich: Lösen eines Gleichungssystems (iterativ, LU -Zerlegung etc.)



Eigenwert-Löser (IV):Jacobi-Davidson-Iteration

$v_k, k = 1, \dots, m$ **orthonormale Vektoren**

$$V_m = [v_1 \mid \dots \mid v_m], \quad W_m = AV_m, \quad H_m = V_m^H W_m$$

Mit extremalem Eigenwert θ and -vektor s von H_m :

Ritz Vektor $u = V_m s$, Residuum $r = W_m s - \theta u$

Neuer Basisvektor t :

$$(I - uu^H)(A - \theta I)(I - uu^H)t \approx -r, \quad t \perp u$$

Orthogonalisiere t mit $V_m \rightarrow v_{m+1}$

Restart : $V_1 = [u], \quad W_1 = AV_1, \quad H_1 = V_1^H W_1$



Eigenwert-Löser (V): Vorkonditionierte JADA-Iteration

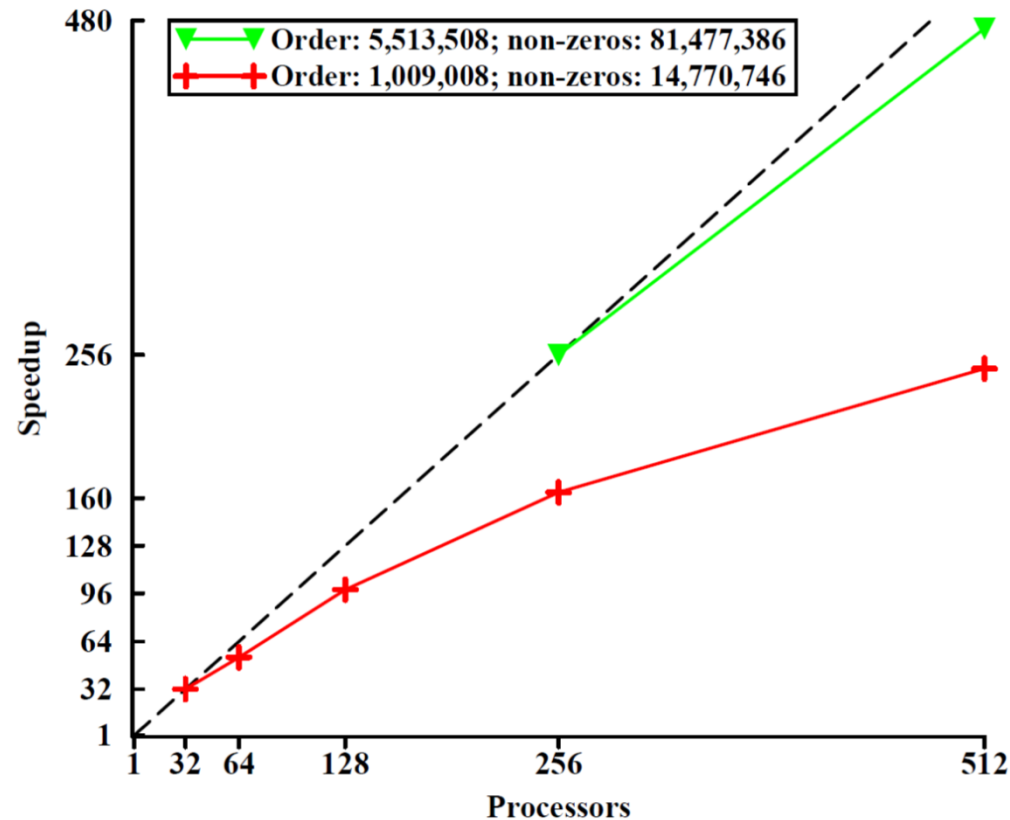
$$(I - uu^H)(A - \theta I)(I - uu^H)t = -r$$

Iterative Gleichungslösung

- Hermitesche Quasi-Minimal Residual method (QMR)
- Hermitesche Transpose-Free QMR (TFQMR)
- Vorkonditionierer ILUT für TFQMR
- **Neu: FGMRES mit DSC-Vorkonditionierung**



Eigenwert-Löser (VI): JADA-Performance



**Elektronen-Phononen-
Wechselwirkung**

**Reelle symmetrische Matrix
4 kleinste Eigenwerte und
Eigenvektoren
Cray T3E (1997)**



Eigenwert-Löser (VII): FEAST-Algorithmus

Eigenschaften

- **Bestimmt alle Eigenwerte und –vektoren in einem vorgegebenen Intervall**
- **Abgeleitet von Krylov-Unterraumverfahren (Arnoldi, Lanczos) und JADA-Techniken**
- **Frei von Orthogonalisierungsoperationen**
- **Hauptoperationen:**
 - **Lösen einiger weniger unabhängiger Gleichungssysteme mit verschiedenen rechten Seiten**
 - **Lösen eines kleinen, reduzierten Eigenwertproblems**



Fragen?

